🔧 ONTBREKENDE BESTANDEN VOOR LIVE DEPLOYMENT

🖥️ BACKEND CORE FILES (KRITIEK)

Server & Configuration

```javascript
// server.js - Main server entry point
// app.js - Express app configuration
// package.json - Dependencies en scripts
// .env.example - Environment template
```

Database Models

```javascript
// models/User.js - User schema
// models/POI.js - Point of Interest model
// models/Analytics.js - Analytics tracking
// models/Conversation.js - Chat history
// models/Booking.js - Reservation system
```

API Routes (ONTBREEKT)

```javascript
// routes/chat.js - Chat API endpoints
// routes/poi.js - POI management
// routes/bookings.js - Reservation system
// routes/recommendations.js - AI recommendations
// routes/admin.js - Admin dashboard
```

Services (KRITIEK ONTBREEKT)

```javascript
// services/WeatherService.js - Weather API integration
// services/POIService.js - POI management logic
// services/EmailService.js - Email notifications
// services/CacheService.js - Redis caching
// services/BookingService.js - Reservation logic
```

Middleware

javascript

```
// middleware/auth.js - JWT authentication
// middleware/rateLimit.js - API rate limiting
// middleware/cors.js - CORS configuration
// middleware/validation.js - Input validation
// middleware/error.js - Error handling
```

📱 MOBILE APP CORE FILES (KRITIEK)

App Configuration

typescript

```
// App.tsx - Main app entry point
// package.json - React Native dependencies
// metro.config.js - Metro bundler config
// babel.config.js - Babel configuration
```

Redux Store

typescript

```
// store/store.ts - Redux store configuration
// store/slices/authSlice.ts - Authentication state
// store/slices/chatSlice.ts - Chat state management
// store/slices/userSlice.ts - User profile state
```

Core Screens (ONTBREEKT)

typescript

```
// screens/chat/ChatScreen.tsx - Main chat interface
// screens/map/MapScreen.tsx - Interactive map
// screens/auth/AuthScreen.tsx - Login/registration
// screens/profile/ProfileScreen.tsx - User profile
// screens/splash/SplashScreen.tsx - App loading
```

API Integration

typescript

```
// services/ApiClient.ts - Backend API client
// services/AuthService.ts - Authentication logic
// services/ChatService.ts - Chat API integration
// services/LocationService.ts - GPS & maps
```

Utils & Theme

typescript

```
// theme/Colors.ts - Mediterranean color palette
// utils/dateUtils.ts - Date formatting
// utils/validation.ts - Form validation
// types/navigation.ts - Navigation types
// types/chat.ts - Chat message types
```

🐳 INFRASTRUCTURE FILES (KRITIEK)

Docker Configuration

yaml

```
# Dockerfile - Container image
# docker-compose.yml - Local development
# docker-compose.prod.yml - Production setup
```

Kubernetes Deployment

yaml

```
# k8s/namespace.yaml - Kubernetes namespace
# k8s/deployment.yaml - App deployment
# k8s/service.yaml - Service configuration
# k8s/ingress.yaml - Load balancer
# k8s/configmap.yaml - Configuration
# k8s/secrets.yaml - Secrets management
```

CI/CD Pipeline

yaml

```
# .github/workflows/deploy.yml - GitHub Actions
# .github/workflows/test.yml - Automated testing
```

Cloud Infrastructure

hcl

```
# terraform/main.tf - AWS infrastructure
# terraform/variables.tf - Configuration variables
# terraform/outputs.tf - Infrastructure outputs
```

🔧 CONFIGURATION FILES

Environment Setup

```bash
# .env.example - Environment template
# .gitignore - Git ignore rules
# .dockerignore - Docker ignore rules
```

Database Setup

```javascript
// database/seeds/pois.js - POI seed data
// database/migrations/ - Database migrations
// database/indexes.js - Database indexes
```

Monitoring & Logging

```yaml
# monitoring/prometheus.yml - Metrics collection
# monitoring/grafana-dashboard.json - Dashboards
# logging/fluentd.conf - Log aggregation
```

🚀 DEPLOYMENT SCRIPTS

Build & Deploy

```bash
# scripts/build.sh - Build automation
# scripts/deploy.sh - Deployment automation
# scripts/migrate.sh - Database migrations
# scripts/seed.sh - Database seeding
```

Health Checks

```javascript
// scripts/health-check.js - Application health
// scripts/load-test.js - Performance testing
```

⚡ PRIORITEIT VOOR LIVE DEPLOYMENT:

🔴 KRITIEK (Zonder deze werkt niets):

1. **server.js** - Backend server entry point
2. **App.tsx** - Mobile app entry point
3. **Database Models** - User, POI, Analytics schemas
4. **API Routes** - Chat, recommendations, authentication
5. **Claude Service Integration** - Volledige Claude API implementatie

🟡 BELANGRIJK (Voor productie kwaliteit):

1. **Docker Configuration** - Container deployment
2. **Kubernetes Files** - Orchestration
3. **Environment Configuration** - .env setup
4. **Database Seeds** - Initial POI data
5. **CI/CD Pipeline** - Automated deployment

🟢 NICE-TO-HAVE (Voor optimalisatie):

1. **Monitoring Setup** - Prometheus/Grafana
2. **Load Testing** - Performance validation
3. **Advanced Security** - WAF, DDoS protection
4. **Analytics Dashboard** - Business intelligence

🎯 CONCLUSIE:

De strategische basis is **EXCELLENT**, maar voor een werkend platform zijn nog **~50 technische implementatiebestanden** nodig. Focus eerst op de KRITIEKE bestanden voor een MVP, dan uitbreiden naar productiekwaliteit.